Extending Lawson's Algorithm to Include the Huber M-Estimator

Iain J. Anderson, John C. Mason,
and Colin Ross

Abstract. When fitting a curve to experimental data, there is no guarantee that the data obtained are as accurate as might be expected. The effect of outside influences may cause the data set to contain outliers. These outliers can have a significant effect on any curve which is fitted to such data. The ℓ_∞-norm, which is particularly appropriate for fitting data with uniformly distributed errors, is extremely sensitive to such outliers, since it minimises the maximum error from the data to the curve. Therefore, a technique which approximates a data set using the ℓ_∞-norm, without being adversely affected by outliers, would be a useful addition to the array of tools available. We present numerical examples to illustrate the use of such a technique and also some practical applications to justify its use.

§1. Introduction

It is widely accepted that the ℓ_∞-norm is the most appropriate measure of the error when approximating data which are very accurate or have errors sampled from a uniform distribution. Unfortunately, because the ℓ_∞ norm is extremely sensitive to outliers, it is not suitable for use in fitting experimental data containing such points. Nevertheless, it may be the case that the ℓ_∞-norm is the most appropriate error measure for the non-outlying data, and so we present an algorithm for finding an ℓ_∞ fit to the non-outliers of a data set.

The algorithm itself is based on a combination of the Huber M-estimator [6] and Lawson's algorithm [7]. There is considerable literature on both techniques as separate subjects, and we mention here only a selection. Lawson's algorithm was first analysed by Lawson [7] in 1961, and was later studied by Rice and Usow [11], Cline [2] and Ellacott [4]. Similarly, the Huber M-estimator was developed by Huber [6] in 1964 and has received considerable attention in the form of algorithms for its solution as well as analyses of its behaviour. Papers by Clark and Osborne [1], Ekblom [3], Madsen and Nielsen

Curve and Surface Fitting: Saint-Malo 1999
Albert Cohen, Christophe Rabut, and Larry L. Schumaker (eds.), pp. 1–8.

ISBN 0-8265-1357-3.

[9], Michelot and Bougeard [10] and Li [8] all look at the Huber M-estimator either in its own right or as one of a class of robust estimators.

In this paper, we consider the problem of fitting a function of the linear form $f(x) = \sum_{j=1}^{n} c_j \phi_j(x)$ to a set of data $\{(x_i, y_i)\}_{i=1}^{m}$, where the $\{\phi_j\}$ are a set of basis functions. To this end, we minimise the residuals $r_i = y_i - f(x_i)$. What our algorithm achieves in practice is to obtain an ℓ_∞ fit for those r_i such that $|r_i|$ is less than the Huber parameter γ, say, and effectively to ignore the remaining data.

The circumstances that require such an algorithm occur in practice, particularly in metrology where extremely accurate readings can be obtained (by, for example, a CD reader) but are subject to the occasional outlier (due, for example, to optical effects). These outliers usually only appear in groups of one or two, so they are isolated, which leads to an easier problem than if they appeared in larger groups. Another metrological situation where this algorithm can be applied is in the measurement of a cylinder in an automotive engine where there is approximately 95% very accurate data, and 5% outliers. Naturally, these problems might require a slightly different fitting technique, but this algorithm is a useful starting point from which more general fitting procedures may be developed in future work.

§2. Background

In this section, we discuss some aspects of both Huber estimation and Lawson's algorithm. In the next section we describe how to combine the two techniques to create a new algorithm which satisfies our requirements.

The Huber M-estimator

The Huber M-estimator is based on the Huber function

$$\rho(t) = \begin{cases} t^2/2, & \text{if } |t| \le 1, \\ |t| - 1/2, & \text{if } |t| > 1, \end{cases} \tag{1}$$

introduced by Huber [6] in 1964, and is defined in the following straightforward way:

$$E = \sum_{i=1}^{m} \rho(r_i/\gamma), \tag{2}$$

where r_i is the residual in the ith datum, and γ is the Huber threshold defining the distinction between "accurate" and "inaccurate" data.

There are several algorithms to solve the problem of minimising (2) with respect to $\mathbf{c}$, several of which are described by Li [8]. However in this paper, we limit ourselves to the Newton method. This involves solving [8]

$$\frac{1}{\gamma^2} A^T D A \mathbf{p} = \frac{1}{\gamma} A^T \mathbf{v}$$

at each iteration, where A is the design matrix with entries $A_{ij} = \phi_j(x_i)$, D is a diagonal matrix with entries $D_{ii} = 1$ if $|r_i| \le \gamma$ and $D_{ii} = 0$ if $|r_i| > \gamma$,

and $\mathbf{v}$ has entries $v_i = \rho'(r_i/\gamma)$. Solving this system gives an update vector $\mathbf{p}$ which should provide a better estimate $\mathbf{c} + \mathbf{p}$ of the solution parameters $\mathbf{c}^*$. In order to ensure convergence, we also incorporate a line search which involves finding a scalar α which is the solution to the equation

$$(A\mathbf{p})^T \rho' \left(\frac{\mathbf{r} + \alpha A\mathbf{p}}{\gamma} \right) = 0.$$

Having found α, we then obtain a new estimate of $\mathbf{c}^*$ by setting $\mathbf{c} := \mathbf{c} + \alpha\mathbf{p}$. We repeat this procedure, updating D and $\mathbf{v}$ as necessary, until we have obtained $\mathbf{c}^*$ to sufficient accuracy.

Weighting

We choose to generalise (2) by introducing weights to obtain a weighted Huber M-estimator of the form $F = \sum_{i=1}^{m} w_i \rho(r_i/\gamma)$, where γ is the Huber threshold, w_i is the weight associated with the ith datum, and r_i is the residual associated with the ith datum. It may be necessary to introduce weights in this way in order to deal with non-identically distributed errors in the data, in which case the weights may be chosen to be the reciprocals of the standard deviations of the underlying probability distributions.

Many algorithms exist to find unweighted Huber fits, and in general, adapting them to find a weighted Huber fit is a straightforward task. As an example, we show how to adapt a Newton-like method.

Weighted Huber algorithm.

1) Calculate $v_i = w_i \rho'(r_i/\gamma)$.
2) If $\frac{1}{\gamma} A^T D_w A\mathbf{p} = -A^T\mathbf{v}$ is consistent, define $\mathbf{p} := -\frac{1}{\gamma}(A^T D_\mathbf{w} A)^+ A^T\mathbf{v}$,
   Otherwise, define $\mathbf{p} := -\frac{1}{\gamma} P^{-1} A^T\mathbf{v}$, where P is a positive definite matrix.
3) Find a steplength $\alpha > 0$ such that $(A\mathbf{p})^T D_\mathbf{w} \rho'((\mathbf{r} + \alpha A\mathbf{p})/\gamma) = 0$.
4) Set $\mathbf{c} := \mathbf{c} + \alpha\mathbf{p}$.

Here, A is the $m \times n$ matrix representing the underlying linear model, $D_\mathbf{w}$ is a diagonal matrix with entries

$$(D_\mathbf{w})_{ii} = \begin{cases} w_i, & \text{if } |r_i/\gamma| \leq 1, \\ 0, & \text{if } |r_i/\gamma| > 1. \end{cases}$$

P is usually the identity matrix, I and Y^+ denotes the pseudo-inverse of a matrix Y, defined so that Y^+ is that matrix X of the same dimensions as Y^T such that $YXY = Y$, $XYX = X$ and YX and XY are symmetric.

We note here that there are many other algorithms for finding a Huber fit, and that most, if not all, can be adapted just as easily.

Lawson's algorithm

This algorithm, analysed by Lawson [7] in 1961, enables an ℓ_∞ fit to be obtained by repeated weighted ℓ_2 fits. The algorithm itself is very straightforward, and involves updating the weights at each iteration according to

$$w_i^{(l+1)} := \frac{w_i G(r_i^{(l)})}{\sum_{k=1}^{m} w_k G(r_k^{(l)})}, \tag{3}$$

where $G(t) = |t|$. The denominator is a normalisation term to ensure that the weights sum to unity. The numerator has the effect of weighting data with large residuals more heavily, with the result that, in the limit, only those data with a maximal residual will have any weight attached to them.

Lawson's algorithm finds the points of extreme oscillation and weights these accordingly to obtain the best ℓ_∞ approximation. The other weights are not important, and in fact converge to zero.

Initial values for the weights are usually chosen to be $w_i^{(1)} = 1/m$, as this treats all the data equally and satisfies the condition that the sum of the weights must be unity. Proofs of convergence require that the $\{\phi_i(x)\}$ form a Chebyshev set, but experimental results (see, for example, [4]) suggest that the algorithm is more generally applicable.

A summary of Lawson's algorithm.

1) Set all weights equal (with the sum of weights equal to unity).
2) Perform a weighted least-squares fit.
3) Calculate the residuals from the weighted least-squares fit.
4) Update the weights according to Lawson's formula (3).
5) Return to Step 2 until convergence is obtained.

§3. The Algorithm

We are concerned with the solution of the problem

$$\min_{\mathbf{c}} \max_{\{r_i : |r_i| \le \gamma\}} |r_i|,$$

where r_i is the residual for the datum (x_i, y_i), and γ is the Huber threshold value. In order to solve this problem, we reformulate it as

$$\min_{\mathbf{c}} \sum_{i=1}^{m} w_i \rho(r_i/\gamma),$$

where ρ is defined as in equation (1), and we adopt an iterative procedure to find $\mathbf{c}$ by performing successive weighted Huber fits. The weights are updated after each iteration in a manner similar to Lawson's original algorithm. While Lawson's algorithm is concerned with finding a minimax fit via a sequence

of weighted least-squares fits, this new algorithm finds a minimax fit to the non-outlying data via a sequence of weighted Huber fits.

Unfortunately, Lawson's rule for updating the weights cannot be used in this new algorithm since the rule would weight the outliers too heavily. As a result, the outliers would be fitted more accurately at the next iteration. The essential point of Lawson's update is to weight those datum points which correspond to the maximal errors of the minimax fit. To maintain this general trend, we need an update in which the function G in (3), rather than being monotonic, instead increases to a peak and then decays, with the peak corresponding to the residual with the largest magnitude which does not exceed γ. The latter is termed the "γ-maximal residual" and denoted by γ_{MR}.

The function we choose in place of $|t|$ is a negative exponential of the form

$$G^{(l)}(t) = \begin{cases} |t|, & \text{if } |t| \leq \gamma_{MR}, \\ \gamma_{MR} e^{-\frac{1}{\gamma_{MR}}(|t|-\gamma_{MR})}, & \text{if } |t| > \gamma_{MR}, \end{cases}$$

and we update the weights at each iteration according to (3). (Note that $G^{(l)}$ changes with the iteration l.)

For $|t| > \gamma_{MR}$, the γ_{MR} factor in $G^{(l)}(t)$ is needed to ensure continuity at $|r_i^{(k)}| = \gamma_{MR}$ and the $-\gamma_{MR}^{-1}$ term in the exponential is used so that the left and right derivatives of $G_i(t)$ are continuous at γ_{MR}. The reason for this second condition is to ensure that points with residuals just over γ_{MR} and those with residuals just less than γ_{MR} are treated similarly.

§4. Convergence

We have obtained favourable results with this algorithm, provided that certain conditions are met. Firstly, the form of the approximating model needs to be appropriate. For example, trying to approximate a set of data corresponding to a quadratic by a straight line will probably lead to problems, as it is likely that a considerable number of the data will be treated as outliers. Secondly, γ needs to be chosen carefully. If γ is chosen to be too small, then there may be many solutions and it may not be possible to predict to which solution the algorithm will converge — if it converges at all.

We therefore conclude that in order to use this algorithm effectively, we first need to have some details of the problem we are to tackle. If we are unsure as to what sort of model to fit to the data, then γ should be chosen to be larger than we might initially require. If we are unsure what value of γ to choose, then some sort of γ-reduction procedure may be effective for finding an appropriate value. An initial value of γ may be chosen by use of the formula $\gamma = 1.9906 \times \mathbf{median}(|r_i - \mathbf{median}(r_i)|)$ (see, for example, Ross et al, [12]).

The effect of using a Lawson-like update with a non-monotonic factor is to increase the weights at the extrema of the minimax approximation and reduce all other weights, including those of the outliers. In practice, the algorithm produces a minimax approximation to a subset of the data with the aim that this subset should be the non-outlying data. Unfortunately, we have been unable thus far to prove convergence for this algorithm. However, it should

be noted that the convergence rate would be expected to be similar to that of Lawson's original algorithm as they essentially do the same task.

§5. Acceleration Schemes

Although the algorithm as it stands is acceptable for small problems, it nevertheless takes a considerable length of time to achieve relaxed convergence conditions. This is no surprise as one of the shortcomings of Lawson's algorithm is its slowness to converge. More specifically, the convergence of Lawson's algorithm is linear with a ratio of τ^* [11], where

$$\tau^* = \max\left[\tau = \frac{|\mathbf{r}^*|}{\max|\mathbf{r}^*|} : \tau < 1\right],$$

with $\mathbf{r}^*$ defined to be the vector of residuals from the optimal ℓ_∞ fit. In many situations, this ratio can be very close to one, leading to rather slow convergence.

One technique to increase the rate of convergence is to use the fact that, upon convergence, the weights corresponding to non-extremal residuals are zero. Specifically, after a set number of normal iterations to allow the weights to settle a little, we may set $w_i = 0$ if $|r_i| \leq \sigma^2/\|r_i\|_\infty$, where $\sigma = \left[\sum_{i=1}^m w_i^{(k)}(r_i^{(k)})^2\right]^{1/2}$. This latter technique is the one presented by Rice and Usow [10], although Ellacott [4] found that it could cause the algorithm to fail.

Of course in the case of this new algorithm, these schemes cannot be applied directly. We need to compensate for those data which are being treated as outliers, thus this scheme is not valid. If it were possible to find some analogue of σ for this new algorithm, then it may be possible to use that analogue in an acceleration scheme.

§6. Numerical Results

We have tested this algorithm extensively and now present some numerical results to illustrate it. In Figure 1, we show a synthesised data set consisting of 95 points lying close to the polynomial $f(x) = 2x^2 - 3x + 1$ with 5 outliers. Figure 1 also shows the best fitting quadratic polynomial to the data obtained by a least-squares fit, by an ℓ_∞ fit and by the new algorithm presented in this paper. The noise in the data is taken from a uniform distribution on $[-0.1, 0.1]$ and we thus choose $\gamma = 0.1$. Table 1 shows the results from the various fits performed. It is clear that both the ℓ_2 and ℓ_∞ fits are unsuitable and are affected by the outliers. However, the new algorithm succeeds in identifying the outliers and successfully ignoring them. Comparing the results from the new algorithm with those from performing least-squares and minimax fits to the data without outliers, we see that they are much more in agreement. In fact, as we would expect, the new algorithm has generated an almost identical fit to the ℓ_∞ fit on the accurate data.

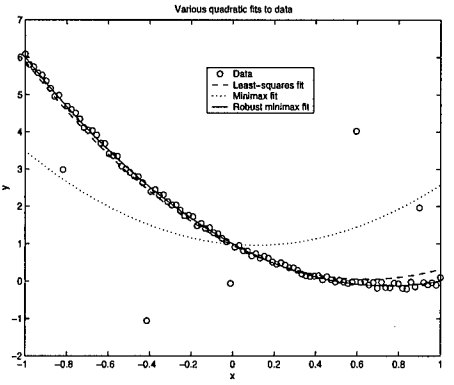

Fig. 1. Various quadratic polynomial fits to a set of data with outliers.

We also note that, while the new algorithm seems to be significantly faster in this example, this is not the case in general. In fact with stricter convergence criteria, Lawson's original algorithm applied to the non-outlying data converges in fewer iterations than the new algorithm. The reason that the minimax fit to the data containing outliers takes fewer iterations is due to the result in Section 5 involving τ^*, which, because of the outliers, is actually quite small ($\tau^* = 0.9497$) compared to $\tau^* = 0.9894$ for the case of the new algorithm.

	ℓ_2	ℓ_∞	New	ℓ_2 (NO)	ℓ_∞ (NO)
c_0	+0.9499	+0.9839	+0.9958	+1.0059	+0.9971
c_1	−2.7868	−0.4571	−2.9951	−3.0007	−3.0001
c_2	+2.1568	+2.0492	+2.0057	+2.0050	+2.0035
Iterations	1	46	39	1	140

Tab. 1. Numerical results: fitting a quadratic (NO : No outliers).

The convergence criterion was the same for both Lawson-like algorithms, namely that the magnitude of the four largest γ-maximal residuals should agree with a relative error of less than 10^{-2}. In addition, no acceleration schemes were used since we needed to obtain a measure of how fast the algorithms were in their unaccelerated form.

§7. Conclusions

We have presented an algorithm for fitting a linear form to data containing uniform noise, contaminated by outliers. Future work will concentrate on three main areas. Firstly, acceleration of the convergence of the algorithm. Secondly, extension to non-linear forms. Thirdly, extension to general ℓ_p norms rather than solely to the ℓ_∞-norm.

References

1. Clark, D. I., and M. R. Osborne, Finite algorithms for Huber's M - estimator, SIAM Journal on Scientific and Statistical Computing **7**(1) (1986), 72–85.
2. Cline, A. K., Rate of convergence of Lawson's algorithm, Mathematics of Computation **26**(117) (1972), 167–176.
3. Ekblom, H., A new algorithm for the Huber estimator in linear models, BIT **28** (1988), 123–132.
4. Ellacott, S. W., *Linear Chebyschef approximation*, M.Sc. thesis, University of Manchester, UK, 1972.
5. Hermey, D., and G. A. Watson, Fitting data with errors in all variables using the Huber M-estimator, SIAM Journal on Scientific Computing **20**(4) (1999), 1276–1298.
6. Huber, P. J., Robust estimation of a location parameter, Annals of Mathematical Statistics **35** (1964), 73–101.
7. Lawson, C. L., *Contributions to the theory of linear least maximum approximation*, Ph.D. thesis, University of California, Los Angeles, CA, USA, 1961.
8. Li, W., Numerical algorithms for the Huber M-estimator problem. In *Approximation Theory VIII*, C. K. Chui and L. L. Schumaker (eds.), World Scientific, New York, NY, USA, 1995, 325–334.
9. Madsen, K., and H. B. Neilsen, Finite algorithms for robust linear regression, BIT **30** (1990), 682–699.
10. Michelot, M. L., and M. L. Bougeard, Duality results and proximal solutions of the Huber M-estimator problem, Applied Mathematics and Optimization **30** (1994), 203–221.
11. Rice, J. R., and K. H. Usow, The Lawson algorithm and extensions, Mathematics of Computation **22** (1968), 118–127.
12. Ross, C., I. J. Anderson, J. C. Mason, and D. A. Turner, Approximating coordinate data that has outliers, in *Advanced Mathematical and Computational Tools in Metrology IV*, P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter (eds.), World Scientific, Singapore, 2000, 210–219.

I. J. Anderson, J. C. Mason, and C. Ross
School of Computing and Mathematics
University of Huddersfield
Queensgate, Huddersfield, West Yorkshire
HD1 3DH, UK

i.j.anderson@hud.ac.uk
j.c.mason@hud.ac.uk
c.ross@hud.ac.uk